

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平10-105467

(43)公開日 平成10年(1998) 4月24日

(51)Int.Cl. ⁶	識別記号	F I
G 0 6 F 12/08	3 2 0	G 0 6 F 12/08 3 2 0
	3 1 0	3 1 0 A
3/06	5 4 0	3/06 5 4 0
12/16	3 2 0	12/16 3 2 0 L

審査請求 未請求 請求項の数16 O L (全 19 頁)

(21)出願番号 特願平9-86171

(22)出願日 平成9年(1997) 4月4日

(31)優先権主張番号 08/630, 906

(32)優先日 1996年4月4日

(33)優先権主張国 米国 (US)

(71)出願人 595026416

シンバイオス・ロジック・インコーポレイ
テッド

アメリカ合衆国 コロラド州 80525 フ
ォート コリンズ ダンフィールド コー
ト 2001

(72)発明者 ロドニー エイ. デコニング

アメリカ合衆国 カンザス州 67226 ウ
ィチタ、ダンベリ 6443

(72)発明者 ドナルド アール. ハムリセク

アメリカ合衆国 カンザス州 67211 ウ
ィチタ、サウス エリー 1702

(74)代理人 弁理士 西山 善章 (外2名)

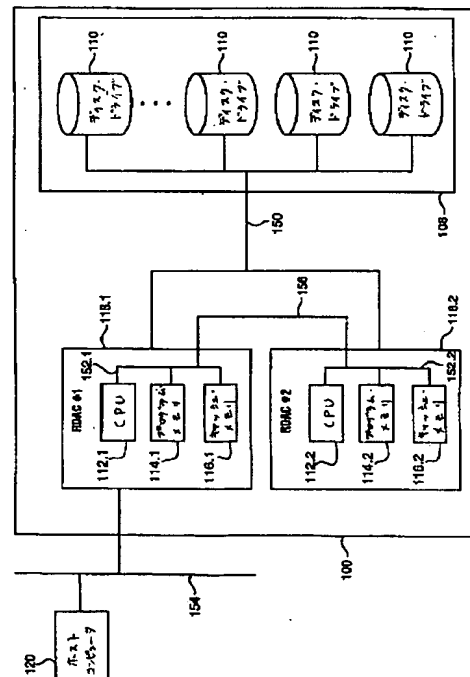
最終頁に続く

(54)【発明の名称】 冗長キャッシュを備えているRAIDコントローラにおけるキャッシュのコンシステンシーを維持するための方法および装置

(57)【要約】 (修正有)

【課題】 RAIDコントローラの冗長ペアの中のキャッシュ・メモリのコンシステンシーを確保する。

【解決手段】 本発明はパワーオン・リセット・サイクル等に応答して、冗長のRAIDコントローラを初期化する。第1のコントローラはホストの要求の処理のために部分的に初期化し、その後、第2のコントローラの部分的初期化を待つ。その待時間の短いタイムアウトの後、あるいはその初期化が間違ったことに応答して、第1のコントローラは第2のコントローラが実質的に初期化するまで、ホスト・コンピュータのI/O要求をキャッシュ動作をバイパスしながら実行するように自分自身を構成する。両方のコントローラが初期化されると、その冗長のキャッシュは同期化されている。その後初期化されたコントローラは正常のミラー型の冗長動作を開始する。



【特許請求の範囲】

【請求項 1】 それぞれにキャッシュ・メモリが付いている冗長ディスク・アレー・コントローラを備えている R A I D 記憶サブシステムにおいて、前記冗長のディスク・アレー・コントローラ間でのキャッシュ・メモリのミラー化のコンシステンシーを維持するための方法であって、

前記冗長の各ディスク・アレー・コントローラの中のキャッシュ・メモリが同期されているかどうかを判定するステップと、

前記のキャッシュ・メモリが同期化されていないという判定に回答して、前記冗長ディスク・アレー・コントローラの少なくとも 1 つをキャッシュ・メモリのライトスルー・モードで動作させるステップと、

前記キャッシュ・メモリが同期化されているという判定に回答して、前記冗長のディスク・アレー・コントローラをキャッシュ・メモリのライトバック・ミラー型のモードで動作させるステップとを含むキャッシュのコンシステンシーを維持するための方法。

【請求項 2】 前記キャッシュ・メモリが同期化されていないという判定に回答して、前記キャッシュ・メモリの 1 つを前記キャッシュ・メモリの他のものにコピーすることによって、前記キャッシュ・メモリを同期化するステップをさらに含んでいる、請求項 1 に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項 3】 前記キャッシュ・メモリのそれぞれに有効性の論理属性指示子があり、その値が「真」である場合、関連のキャッシュ・メモリが不揮発性のままであることを示し、前記判定するステップが、両方のキャッシュ・メモリの有効性論理属性値が「真」である場合に前記キャッシュ・メモリは同期化されていると判定するステップと、前記キャッシュ・メモリのうちの少なくとも 1 つの有効性論理属性値が「偽」である場合に、前記キャッシュ・メモリは同期化されていないと判定するステップとを含んでいる、請求項 1 に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項 4】 前記キャッシュ・メモリが同期化されていないという判定に回答して、キャッシュ・メモリを同期化するステップをさらに含んでいて、該同期化するステップが、両方のキャッシュ・メモリの有効性論理属性値が「偽」と判定するステップと、両方のキャッシュ・メモリの有効性論理属性値が「偽」という判定に回答して、両方のキャッシュ・メモリの内容をパージするステップと、前記キャッシュ・メモリの 1 つの有効性論理属性値が「真」と判定するステップと、有効性論理属性値が「真」である前記キャッシュ・メモリの前記 1 つの内容を前記キャッシュ・メモリの他のも

のに対してコピーするステップと、

前記各キャッシュ・メモリの前記有効論理属性値を

「真」にセットするステップとを含んでいることを特徴とする、請求項 3 に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項 5】 前記キャッシュ・メモリのそれぞれがネイティブ論理属性値を持っていて、その値が「真」であった場合、そのキャッシュ・メモリの中に記憶されている内容が R A I D 記憶サブシステムと最近関係付けられたものであることを示し、前記判定するステップが、両方のキャッシュ・メモリのネイティブ論理属性値が「真」であった場合に、前記キャッシュ・メモリは同期化されていると判定するステップと、

前記キャッシュ・メモリのうちの少なくとも 1 つのネイティブ論理属性値が「偽」であった場合に、前記キャッシュ・メモリは同期化されていないと判定するステップとを含んでいることを特徴とする、請求項 1 に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項 6】 前記キャッシュ・メモリが同期化されていないという判定に回答して、キャッシュ・メモリを同期化するステップをさらに含んでいて、前記同期化するステップが、

両方のキャッシュ・メモリのネイティブ論理属性値が「偽」と判定するステップと、

論理のキャッシュ・メモリのネイティブ論理属性値が「偽」とあるという判定に回答して両方のキャッシュ・メモリの内容をパージするステップと、前記キャッシュ・メモリの 1 つのネイティブ論理属性値が「真」と判定するステップと、

ネイティブ論理属性値が「真」である前記キャッシュ・メモリの前記 1 つの内容を前記キャッシュ・メモリの他のものへコピーするステップと、

前記キャッシュ・メモリのそれぞれの前記ネイティブ論理属性値を「真」に設定するステップとを含んでいることを特徴とする、請求項 5 に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項 7】 前記キャッシュ・メモリのそれぞれが有効性論理属性値を持っていて、その値が「真」である場合、その関連のキャッシュ・メモリが不揮発性のままであり、前記キャッシュ・メモリのそれぞれがネイティブ論理属性値を持っていて、その値が「真」である場合、キャッシュ・メモリの中に記憶されている内容が R A I D 記憶サブシステムと最近関係付けられたことを示し、前記判定するステップが、

両方のキャッシュ・メモリのネイティブ論理属性値が「真」である場合、そして両方のキャッシュ・メモリの有効性論理属性値が「真」である場合に、前記キャッシュ・メモリは同期化されていると判定するステップと、前記キャッシュ・メモリのうちの少なくとも 1 つのネイティブ論理属性値が「偽」である場合、あるいは前記キ

10

20

30

40

50

キャッシュ・メモリのうちの少なくとも1つの有効性論理属性値が「偽」である場合に、前記キャッシュ・メモリは同期化されていないと判定するステップとを含んでいることを特徴とする、請求項1に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項8】 前記キャッシュ・メモリが同期化されていないという判定に応答して、キャッシュ・メモリを同期化するステップをさらに含んでいて、前記同期化するステップが、
ネイティブ論理属性値が「真」であって有効性論理属性値が「真」である前記キャッシュ・メモリがないことを判定するステップと、
ネイティブ論理属性値が「真」であって有効性論理属性値が「真」である前記キャッシュ・メモリがないという判定に応答して、両方のキャッシュ・メモリの内容をパージするステップと、
前記キャッシュ・メモリの1つの論理属性値が「真」であって有効性論理属性値が「真」であることを判定するステップと、
ネイティブ論理属性値が「真」であって有効性論理属性値が「真」である前記キャッシュ・メモリの前記1つの内容を前記キャッシュ・メモリの他のものに対してコピーするステップと、
前記各キャッシュ・メモリの前記有効性論理属性値を「真」に設定するステップとを含んでいることを特徴とする、請求項7に記載のキャッシュのコンシステンシーを維持するための方法。

【請求項9】 それぞれがキャッシュ・メモリを備えている冗長のディスク・アレー・コントローラを備えているRAID記憶サブシステムにおいて、前記冗長のディスク・アレー・コントローラの内部で前記冗長のディスク・アレー・コントローラ間のキャッシュ・メモリのミラー化のコンシステンシーを維持する装置であって、該装置が、
前記冗長のディスク・アレー・コントローラのそれぞれの中のキャッシュ・メモリが同期化されているかどうかを判定するための手段と、
前記キャッシュ・メモリが同期化されていないという判定に応答して、前記冗長のディスク・アレー・コントローラの少なくとも1つをキャッシュ・メモリのライトスルー・モードで動作させるための手段と、
前記キャッシュ・メモリが同期化されているという判定に応答して、前記冗長のディスク・アレー・コントローラをキャッシュ・メモリのライトバック・ミラー型モードで動作させるための手段とを備えてなるキャッシュのコンシステンシーを維持するための装置。

【請求項10】 前記キャッシュ・メモリが同期化されていないという判定に応答して、前記キャッシュ・メモリの1つを前記キャッシュ・メモリの他のものへコピーすることによって、前記キャッシュ・メモリを同期化す

るための手段をさらに備えてなる、請求項9に記載のキャッシュのコンシステンシーを維持するための装置。

【請求項11】 前記キャッシュ・メモリのそれぞれに有効性の論理属性指示子があり、その値が「真」である場合、関連のキャッシュ・メモリが不揮発性のままであることを示し、前記判定するための手段が、
両方のキャッシュ・メモリの有効性論理属性値が「真」である場合に前記キャッシュ・メモリは同期化されていると判定するための手段と、

10 前記キャッシュ・メモリのうちの少なくとも1つの有効性論理属性値が「偽」である場合に、前記キャッシュ・メモリは同期化されていないと判定するための手段とを備えてなる、請求項9に記載のキャッシュのコンシステンシーを維持するための装置。

【請求項12】 前記キャッシュ・メモリが同期化されていないという判定に応答して、キャッシュ・メモリを同期化するための手段をさらに備えていて、該同期化するための手段が、

両方のキャッシュ・メモリの有効性論理属性値が「偽」であると判定するための手段と、
20 両方のキャッシュ・メモリの有効性論理属性値が「偽」であるという判定に応答して、両方のキャッシュ・メモリの内容をパージするための手段と、

前記キャッシュ・メモリの1つの有効性論理属性値が「真」であると判定するための手段と、
有効性論理属性値が「真」である前記キャッシュ・メモリの前記1つの内容を前記キャッシュ・メモリの他のものに対してコピーするための手段と、

30 前記各キャッシュ・メモリの前記有効性論理属性値を「真」にセットするための手段とを備えてなることを特徴とする、請求項11に記載のキャッシュのコンシステンシーを維持するための装置。

【請求項13】 前記キャッシュ・メモリのそれぞれがネイティブ論理属性値を持っていて、その値が「真」であった場合、それはそのキャッシュ・メモリの中に記憶されている内容がRAID記憶サブシステムと最近関係付けられたものであることを示し、前記判定するための手段が、

両方のキャッシュ・メモリのネイティブ論理属性値が「真」であった場合に、前記キャッシュ・メモリは同期化されていると判定するための手段と、

40 前記キャッシュ・メモリのうちの少なくとも1つのネイティブ論理属性値が「偽」であった場合に、前記キャッシュ・メモリは同期化されていないと判定するための手段とを備えてなることを特徴とする、請求項9に記載のキャッシュのコンシステンシーを維持するための装置。

【請求項14】 前記キャッシュ・メモリが同期化されていないという判定に応答して、キャッシュ・メモリを同期化するための手段をさらに備えていて、該同期化するための手段が、

両方のキャッシュ・メモリのネイティブ論理属性値が「偽」であると判定するための手段と、
 論理のキャッシュ・メモリのネイティブ論理属性値が「偽」であるという判定に回答して両方のキャッシュ・メモリの内容をパージするための手段と、
 前記キャッシュ・メモリの1つのネイティブ論理属性値が「真」とであると判定するための手段と、
 ネイティブ論理属性値が「真」である前記キャッシュ・メモリの前記1つの内容を前記キャッシュ・メモリの他のものへコピーするための手段と、
 前記キャッシュ・メモリのそれぞれの前記ネイティブ論理属性値を「真」に設定するための手段とを備えてなることを特徴とする、請求項13に記載のキャッシュのコンシステンシーを維持するための装置。
 【請求項15】 前記キャッシュ・メモリのそれぞれが有効性論理属性値を持っていて、その値が「真」である場合、その関連のキャッシュ・メモリが不揮発性のままであり、前記キャッシュ・メモリのそれぞれがネイティブ論理属性値を持っていて、その値が「真」である場合、キャッシュ・メモリの中に記憶されている内容がRAID記憶サブシステムと最近関係付けられたことを示し、前記判定するための手段が、
 両方のキャッシュ・メモリのネイティブ論理属性値が「真」である場合、そして両方のキャッシュ・メモリの有効性論理属性値が「真」である場合に、前記キャッシュ・メモリは同期化されていると判定するための手段と、
 前記キャッシュ・メモリのうちの少なくとも1つのネイティブ論理属性値が「偽」である場合、あるいは前記キャッシュ・メモリのうちの少なくとも1つの有効性論理属性値が「偽」である場合に、前記キャッシュ・メモリは同期化されていないと判定するための手段とを備えてなることを特徴とする、請求項9に記載のキャッシュのコンシステンシーを維持するための装置。
 【請求項16】 前記キャッシュ・メモリが同期化されていないという判定に回答して、キャッシュ・メモリを同期化するための手段をさらに備えていて、該同期化するための手段が、
 ネイティブ論理属性値が「真」であって有効性論理属性値が「真」である前記キャッシュ・メモリがないことを判定するための手段と、
 ネイティブ論理属性値が「真」であって有効性論理属性値が「真」である前記キャッシュ・メモリがないという判定に回答して、両方のキャッシュ・メモリの内容をパージするための手段と、
 前記キャッシュ・メモリの1つの論理属性値が「真」であって有効性論理属性値が「真」であることを判定するための手段と、
 ネイティブ論理属性値が「真」であって有効性論理属性値が「真」である前記キャッシュ・メモリの前記1つの

内容を前記キャッシュ・メモリの他のものに対してコピーするための手段と、
 前記各キャッシュ・メモリの前記有効性論理属性値を「真」に設定するための手段とを備えてなることを特徴とする、請求項15に記載のキャッシュのコンシステンシーを維持するための装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、概して、ディスク・アレー・サブシステム(RAID)の内部で動作する制御方法に関し、特にキャッシュのリセット・サイクルを通じて冗長のキャッシュ間でのキャッシュのミラーリング(mirroring)のコンシステンシーを維持するための方法に関する。

【0002】

【従来の技術、及び、発明が解決しようとする課題】現代の大容量記憶サブシステムはホスト・コンピュータ・システムのアプリケーションからのユーザ需要を満たすために増大する記憶容量を提供し続けている。この大容量記憶装置に頼る度合いが大きいために、その信頼性の向上に対する要求も高い。大容量記憶サブシステムの信頼性を維持または向上させながら、より大きな記憶容量に対する需要に応えるために、各種の記憶装置の構成および方式が普通適用されている。

【0003】記憶容量および信頼性の向上のためのこれらの大容量記憶の需要に対する一般的な解決策は、種々の故障の場合にデータの完全性を確保するために記憶データの冗長性を許す幾何学的配置に構成された複数の小容量記憶モジュールを使うことである。多くのそのような冗長のサブシステムにおいて、多くの普通の故障からの回復はデータの冗長性、誤り符号、およびいわゆる「ホット・スペア」(以前にアクティブであった記憶モジュールの故障したものと置き換えるためにアクティブにすることができる余分の記憶モジュール)を使用するように、その記憶サブシステムそのものの内部で自動化することができる。これらのサブシステムは通常安価な(または独立の)ディスクの冗長のアレーとして(あるいは、より一般には頭字語RAID)と呼ばれている。「安価なディスクの冗長の配列(RAID)の場合」と題する、カリフォルニア大学バークレー校からデビッドA. パターソン他によって1987年に出版された本にRAID技術の基本概念がレビューされている。

【0004】パターソンの出版物の中では5つの「レベル」の標準の幾何学的配置が定義されている。最も単純なアレー、すなわち、RAIDレベル1のシステムはデータを記憶するための1台またはそれ以上のディスクと、そのデータ・ディスクに書き込まれる情報のコピーを記憶するための同数の追加の「ミラー」ディスクを含んでいる。残りのRAIDレベル、すなわち、RAIDレベル2、3、4および5のシステムとして識別される

レベルでは、数台のデータ・ディスクにまたがって記憶のための場所にデータをセグメント化する。誤りチェックまたはパリティ情報を格納するために1台またはそれ以上の追加のディスクが利用される。

【0005】RAID記憶サブシステムは通常はその冗長のアレーの管理の詳細をユーザまたはホスト・システムから遮蔽する制御モジュールを利用する。そのコントローラによって、そのサブシステムはホスト・コンピュータに対して1つの単独の、高信頼の、大容量ディスク・ドライブとして見える。実際には、そのRAIDコントローラはホスト・コンピュータ・システムから供給されるデータを、冗長性および誤りチェック情報を備えた複数の小型の独立のドライブにまたがって分散させ、サブシステムの信頼性を向上させることができる。RAIDサブシステムはそのRAIDサブシステムの性能をさらに改善するために大容量のキャッシュ・メモリ構造を提供することが多い。そのキャッシュ・メモリはディスク・アレー上の記憶ブロックがキャッシュの中のブロックにマップされるように制御モジュールに関連付けられている。また、このマッピングはホスト・システムに対してはトランスペアレント（transparent）である。ホスト・システムは単純にデータのブロックの読み書きを要求し、RAIDコントローラはそのディスク・アレーおよびキャッシュ・メモリを必要に応じて操作する。

【0006】信頼性をさらに改善する目的で、制御電子回路の故障によるサブシステムの故障率を減らすために冗長の制御モジュールを用意することがこの分野の技術において知られている。いくつかの冗長のアーキテクチャにおいては、制御モジュールのペアがディスク・ドライブの同じ物理的アレーを制御するように構成されている。1つのキャッシュ・メモリ・モジュールがその制御モジュールの各冗長ペアに関連付けられている。その冗長の制御モジュールは互いに通信し、キャッシュ・モジュールが同期化されることを確保する。以前の設計においては、制御モジュールの冗長ペアはそれぞれのパワー・オン初期化時に（あるいはリセット操作の後で）通信する。キャッシュ・モジュールの同期化を確保するために、それぞれの通信をその冗長の制御モジュールが完了するまでの間は、RAID記憶サブシステムはホスト・コンピュータの要求を完了することに関しては利用できなくなる。キャッシュ・モジュールが「同期はずれ」であることが分かった場合、同期を回復するために必要な時間はかなりな長さになり得る。さらに、制御モジュールの冗長ペアの1つが故障した場合、そのRAID記憶サブシステムが利用できなくなる時間がさらに延びることになる。そのRAIDサブシステムがホスト・コンピュータの要求の処理を開始するために、故障している冗長の制御モジュールを交換するための手動の（オペレータによる）介入が必要となる。

【0007】上記の観点において、ディスク・ドライブに対するキャッシュされた書き込みを知らせるためのRAIDサブシステムについての必要なオーバーヘッド処理をさらに減らす、RAIDサブシステムに対する改善されたキャッシュ・アーキテクチャおよび制御方法に対するニーズが存在することは明らかである。

【0008】

【課題を解決するための手段】本発明は上記および他の問題点を解決し、それによって、ホスト・コンピュータのI/O要求を処理しながら冗長のキャッシュのコンシステンシーを確保するために、RAID記憶サブシステムが制御モジュールの冗長ペアを備えることができる方法および関連の装置を提供することにより、有用な技術を進歩させる。特に、本発明の方法および装置は2台の冗長制御モジュールのうちの第1の「チェック・イン」に続いて第2の冗長の制御モジュールの「チェック・イン」を提供する。本発明の方法のチェックイン処理は、そのコントローラ・ペアがミラー動作を開始する前に、その冗長キャッシュ・モジュールが同期化されることを確保する。しかし、従来の設計と違って、本発明の方法は第2のコントローラが正常にチェック・インする前に、第1のコントローラがホスト・コンピュータのI/O要求を処理できるようにもする。次に、本発明の方法は第1のコントローラによるホスト・コンピュータのI/O要求の処理の間に冗長キャッシュのコンシステンシーを確保することによって、第2のコントローラの「レイト・チェック・イン（late check-in）」を調整する。

【0009】特に、本発明は2つの冗長制御モジュールの最初のものを「チェック・イン」するために1日のうちのパワー・オン・リセット・サイクル（または、任意の他の同様なリセット機能）の開始時に必要な処理を実行する。「チェック・イン」プロセスは制御モジュールのキャッシュ・モジュールを検査して、その内容がRAID記憶サブシステムと同期化されているかどうかを判定する。まず最初に、チェック・イン・プロセスはミラー型の動作（本発明のRAIDサブシステムの構成によって示されているような）が必要であるかどうかを決定する。ミラー型の動作が当面は不要であった場合、第2のコントローラと同期化させるための処理はそれ以上必要ではない。第1のコントローラは第2のコントローラとそのキャッシュ・メモリを同期化することを顧慮せずに独立に機能することができる。ミラー型の動作が現在望まれていると仮定して、第1のコントローラは自分がそのチェック・イン処理を待っていることを第2のコントローラに対して知らせる。

【0010】第1のコントローラからの信号に応答して、第2のコントローラは次に自分のキャッシュ・モジュールをコンシステンシーに関してチェックすることによってチェック・インする。キャッシュ・モジュールの

コンシステンシー（第1または第2の制御モジュールのいずれかにおける）は、その制御モジュールが現在のRAIDサブシステムにおいて最近使われたこと、そしてキャッシュ・メモリの不揮発性を維持しているバッテリー・サブシステムが正しく動作していることをチェックすることによって判定される。現在のRAIDサブシステムにおいて最近に使われた制御モジュールは、ここでは「ネイティブ（native）」と呼ばれ、一方、現在のRAIDサブシステムにおいて最近に使われていなかった制御モジュールは「フォーレン（foreign）」制御モジュールと呼ばれる。制御モジュールのキャッシュ・メモリに関連しているバッテリー・サブシステムが正しく動作している場合、そのキャッシュ・メモリはRAIDサブシステムにおいて最近動作させられた時と同じ内容を持っていることが確保される。両方の制御モジュールがチェック・インされ、そしてそれらが両方ともネイティブであって、そして両方のバッテリー・サブシステムが正しく動作している場合、その冗長制御モジュールのペアはミラー型のキャッシュ動作を継続することができる。制御モジュールのうちの1つがフォーレン状態を示すか、あるいは「不良」のバッテリー・サブシステムのために無効なキャッシュを示している場合、第1の制御モジュールはその2つのキャッシュ・メモリ・モジュールが再び同期化されて復元されるまで、ホスト・コンピュータのI/O要求を「ライトスルー（write-through）」モードで実行し続ける（キャッシュ・メモリの使用をバイパスして）。ライトスルー・モードで第1のコントローラによってホスト・コンピュータのI/O要求が処理されている時にバックグラウンドにおいて適切なコピー動作を行なうことにより、キャッシュが同期化される。いずれのキャッシュも有効でない場合（例えば、両方のコントローラのキャッシュの内容がそれぞれフォーレン状態のためにキャッシュの内容が無効になっているか、あるいはバッテリー・サブシステムが動作しない場合）、両方の制御モジュールに対するキャッシュがページされ（内容が捨てられ）、そして冗長の制御モジュール・ペアにおいて空のキャッシュでスタートすることによってミラー型の動作が継続する。

【0011】第1のコントローラがチェックされて動作できる状態になると、それは第2のコントローラがチェック・インするための短い期間だけ待ってからキャッシュの同期をチェックする。短いタイムアウトの後、第1のコントローラは第2のコントローラの「レイト・チェック・イン」を待っている時に「ライトスルー」の動作へ切り替わる。本発明のこの機能によって、RAIDサブシステムは制御モジュールの冗長ペアがそれぞれのキャッシュを同期化している間、ホスト・コンピュータのI/O要求を処理することができる（指定されたモードの動作によって）。

【0012】従って、本発明の目的は、ディスク・アレー・コントローラの冗長ペアの中のキャッシュ・メモリのコンシステンシーを確保するための方法および関連の装置を提供することである。

【0013】本発明の他の目的は、ホスト・コンピュータのI/O要求の処理と並行して、ディスク・アレー・コントローラの冗長ペアにおけるキャッシュ・メモリのコンシステンシーを確保するための方法および関連の装置を提供することである。

【0014】本発明のさらに他の目的は、各ペアの第1の制御モジュールがホスト・コンピュータのI/O要求を処理し続けている間に、キャッシュされた制御モジュールの冗長ペアを同期化するための方法および関連の装置を提供することである。

【0015】本発明の上記および他の目的、態様、特徴および利点が次の記述および付属の図面に従って明らかとなる。

【0016】

【発明の実施の形態】本発明は各種の変更および他の形式が可能であるが、その特定の実施例が図面の中の例の方法によって示されており、その中で詳細に説明される。しかし、それは本発明をその開示された特定の形式に限定することを意図するものではなく、逆に、本発明は付属の特許請求の範囲によって定義されているような本発明の精神および範囲内に入るすべての変更、等価物、および代替物をカバーする。

【0017】＜RAIDの概要＞図1は冗長のディスク・アレー・コントローラ118、1および118、2（以下、RDACと呼ばれる）を備えている代表的なRAID記憶サブシステム100のブロック図であり、その中で本発明の方法および関連の装置を適用することができる。RAID記憶サブシステム100は少なくとも1つのRDAC 118、1および118、2を含んでいる。118、1および118、2の各RDACはバス（または、複数のバス）150を経由してディスク・アレー108に接続されており、そしてバス154を経由してホスト・コンピュータ120に接続されている。ディスク・アレー108は複数のディスク・ドライブ110から構成されている。この分野の技術に普通に習熟している人であれば、RDAC118、1および118、2とディスク・アレー108（ディスク・ドライブ110を含んでいる）との間のインターフェース・バス150は、SCSI、IDE、EIDE、IPI、ファイバ・チャネル、SSA、PCIなどのいくつかの業界標準インターフェース・バスの任意のものが使えることは容易に分かる。バス150を制御するのに適しているRDAC 118、1および118、2の内部の回路（図示せず）はこの分野の技術に普通に習熟している人にはよく知られている。RDAC 118、1および118、2とホスト・コンピュータ120との間のインターフェ

ース・バス154としてはSCSI、イーサネット（LAN）、トークンリング（LAN）などのいくつかの任意の業界標準のインターフェース・バスを使うことができる。RDAC 118. 1および118. 2の内部でバス154を制御するのに適している回路（図示せず）はこの分野の技術に普通に習熟している人にとってはよく知られている。

【0018】図1に示されているように、RAID記憶サブシステム100はよく知られているRAIDレベル（例えば、レベル0～5）の任意のレベルを実装するために利用することができる。各種のRAIDレベルが、ディスク・アレー108の中のディスク・ドライブ110を、関連付けられているRAIDコントローラが論理的に細分する、すなわち区画化する方法によって区別される。例えば、RAIDレベル1の機能を実装する時、ディスク・アレー108のディスク・ドライブ110の適当な半分がデータの記憶および検索のために使われ、一方、他の半分は最初の半分のデータ記憶内容をミラーするためにRAIDコントローラによって駆動される。さらに、RAIDのレベル4の機能を実装している時、そのRAIDコントローラはディスク・アレー108の中のディスク・ドライブ110の一部分をデータの記憶のために利用し、そして残りのディスク・ドライブ110を誤りチェック／訂正用情報（例えば、パリティ情報）の記憶のために利用する。以下に説明されるように、本発明の方法および関連の装置は標準のRAIDレベルの任意のものと結合してRAID記憶サブシステム100に適用することができる。

【0019】RDAC 118. 1はCPU 112. 1、プログラム・メモリ114. 1（例えば、CPU 112. 1の動作のためにプログラムの命令および変数を記憶するためのROM/RAMのデバイス）、およびディスク・アレー108の中に記憶されているデータに関連付けられているデータおよび制御情報を記憶するためのキャッシュ・メモリ116. 1を含んでいる。CPU 112. 1、プログラム・メモリ114. 1、およびキャッシュ・メモリ116. 1はメモリ・バス152. 1を経由して接続され、CPU 112. 1がそのメモリ・デバイスの中に情報を記憶し、そして検索できるようにしている。本発明のデータ構造はキャッシュ・メモリ116. 1の中に組み込まれ、CPU 112. 1の内部で動作する手段によって生成され、そして操作される。RDAC 118. 2はRDAC 118. 1と同じものであり、CPU 112. 2、プログラム・メモリ114. 2およびキャッシュ・メモリ116. 2から構成され、それらはすべてメモリ・バス152. 2を経由して接続されている。各RDACが互いに通信できるようにするために、RDAC 118. 1および118. 2は共有バス156を経由して接続されている。RDAC 118. 1および118. 2はRAIDサブ

システム100の内部で交換可能な装置であり、故障のRDACのホット・スワップを含む簡単な置き換えが可能である。この分野の技術に普通に習熟している人であれば、図1のブロック図が本発明を実現することができる単なる設計の一例を意図していることは容易に分かる。多くの代替のコントローラおよびサブシステムの設計によって、本発明の方法および関連の装置および構造を実現することができる。

【0020】＜冗長キャッシュのアーキテクチャ＞118. 1または118. 2の各RDACの中の各CPU 112. 1または112. 2は、共有バス156を経由して他のRDACのキャッシュ・メモリ116. 1または116. 2を操作することができる。このRDACはCPU 112. 1および112. 2の中で動作するソフトウェアおよび制御方法によって変わるいくつかのモードで使うことができる。デュアル・アクティブのRDACペア・モードの操作においては、各キャッシュ・メモリ116. 1および116. 2はCPU 112. 1および112. 2の内部で動作する制御方法によって、対応しているCPU（それぞれのメモリ・バス152. 1および152. 2を通して付加されている）によって使われるための第1のセクションと、代替のRDACによって使われるための第2のセクション（共有バス156による）とに論理的に分けられている。図2に示されているように、RDAC 118. 1のキャッシュ・メモリ116. 1は「MY__CACHE」と名付けられている第1のセクションを備えており、このセクションはディスク・ドライブとの間のI/O要求をバッファするためにバス152. 1を経由してCPU 112. 1によって使われる。「ALT__CACHE」と名付けられているキャッシュ・メモリ116. 1の第2の部分は共有バス156を経由して代替のRDACモジュール、すなわち、118. 2によって使われるために予約されている。同様に、RDAC 118. 2の中のキャッシュ・メモリ116. 2はバス152. 2を経由してCPU 112. 2によって使われるための第1のセクション「MY__CACHE」と、共有バス156を経由して代替のRDAC 118. 1によって操作されるための第2のセクション「ALT__CACHE」に論理的に分割されている。

【0021】デュアル・アクティブのRDACペア・モードにおいては、RDACの各ペア118. 1および118. 2は代替のRDACのキャッシュ以外に、自分自身のキャッシュの中のキャッシュ情報を維持するために他と並行してアクティブになっている。各RDACはそれ自身の特定の論理ユニット（ディスク・アレー108の中のディスク・ドライブ・グループはここではLUNとも呼ばれる）を制御することができる。詳細には、RDAC 118. 1はキャッシュ・メモリ116. 1の第1のセクション、MY__CACHEの中の、そして、

キャッシュ・メモリ116. 2の第2のセクション、ALT_CACHEの中のその論理ユニットの管理に関連したキャッシュ情報を維持している。逆に、RDAC 118. 2はキャッシュ・メモリ116. 2の第1のセクション、MY_CACHEと、キャッシュ・メモリ116. 1の第2のセクション、ALT_CACHEの中のその論理ユニットの管理に関連したキャッシュ情報を維持している。このモードにおいては、そのペアの各RDACは自分自身およびそのペアになっているRDAC（代替のRDAC）によって操作される現在のキャッシュ情報の完全なスナップショットを備えている。1つのRDACが初期化される時、キャッシュ・メモリの内容が無効である1つのRDACは、自分のキャッシュ・メモリ（自分自身のMY_CACHEセクション）を代替のRDACのキャッシュ・メモリ（代替のALT_CACHEセクション）から更新することができる。同様に、代替のRDAC（キャッシュ・メモリの内容が有効であるRDAC）は、自分の有効なキャッシュ・メモリ（自分のMY_CACHEセクション）を無効なALT_CACHEへコピーすることによって、その無効なキャッシュ・メモリのALT_CACHEセクションを更新することができる。以下に詳細に説明される本発明の方法は、RDACペアのキャッシュ・メモリの同期化を確保するために、このキャッシュ更新手順のシーケンスを制御し管理する。

【0022】アクティブ・パッシブのRDACペア・モードの動作においては、RDAC 118. 1または118. 2のうちのいずれか1つはそれが自分のキャッシュ・メモリを維持するためにホスト・コンピュータのI/O要求を処理するように「アクティブ」であるとみなされ、一方、他のRDACはそのアクティブRDACの中のキャッシュ・メモリのコピーを自分のキャッシュ・メモリの中に単純に維持しているという意味で「パッシブ」である。そのパッシブのRDACはホスト・コンピュータのI/O要求を処理しない。代わりに、アクティブのRDACがホスト・コンピュータのI/O要求を処理し、自分自身のキャッシュ・メモリをそれに従って更新し、そして代替のRDACのキャッシュ・メモリの中に自分のキャッシュ・メモリのミラー・イメージを維持する。図3に示されているように、アクティブであるRDAC 118. 1はホスト・コンピュータのI/O要求を処理し、それに従ってキャッシュ・メモリ 116. 1を更新することによって、自分のキャッシュ・メモリ116. 1「ALT_CACHE」を維持する。CPU 112. 1はバス152. 1を経由してアクティブのキャッシュ・メモリ116. 1を更新する時、それは共有バス156を経由してパッシブのRDAC 118. 2の中のパッシブ・キャッシュ・メモリ116. 2に対してキャッシュの変更をミラーすることも行なう。

【0023】アクティブ・パッシブのRDACペア・モ

ードの動作において、RDAC 118. 1および118. 2に関連付けられているキャッシュ・メモリ116. 1および116. 2はアクティブなRDACによって維持される。これによってそのペアのパッシブのRDACが、一次RDACの故障がセンスされた時にただちに制御を確保することができる。同様に、RDACペアが初期化される時、そのRDACは2つのRDACのキャッシュ・メモリを1つから別のRDACへコピーすることによって、同期化することができる。例えば、アクティブのRDAC 118. 1のキャッシュ・メモリ116. 1が無効であった場合、パッシブのRDAC 118. 2の有効なキャッシュ・メモリ116. 2の内容をアクティブのRDAC 118. 1のキャッシュ・メモリ116. 1へコピーすることができる。逆に、そのRDACペアが初期化される時にパッシブのRDAC 118. 2のキャッシュ・メモリ116. 2が無効であった場合、アクティブのRDAC 118. 1の有効なキャッシュ・メモリ116. 1の内容をパッシブのRDAC 118. 2のキャッシュ・メモリ116. 2へコピーすることができる。以下に説明される本発明の方法はRDACペアのキャッシュ・メモリの同期化を確保するためにこのキャッシュの更新手順のシーケンスを制御し、管理する。冗長のペアのコントローラのすべての動作モードにおいて、キャッシュ・メモリはそのキャッシュ・メモリの内容を特定のRAIDサブシステムに関連付けるシグネチャー・データを含んでいる。そのシグネチャー情報のマッチによって判定されるような、そのキャッシュ・メモリが動作しているRAIDサブシステムに内容が関連付けられているキャッシュ・メモリを持っているRDACは、ここでは「ネイティブ」コントローラと呼ばれる。逆に、シグネチャー情報のミスマッチによって判定されるように、そのキャッシュ・メモリが現在動作しているRAIDサブシステムに内容が関連付けられていないキャッシュ・メモリを持っているRDACは、ここでは「フォーレン (foreign)」コントローラと呼ばれる。

【0024】さらに、上記のように、各RDACのキャッシュ・メモリはバッテリー・サブシステム、または他のよく知られている装置を含んでいて、そのRDACモジュールに対する電源の停電によるキャッシュ・メモリに関連付けられる有効性（非有効性）を維持している。そのバッテリー・サブシステムは、そのバッテリー・サブシステムが或る時点で故障したことがあり、従ってその関連付けられたキャッシュ・メモリ・サブシステムの非有効性について疑問を投げ掛けていることを示すためのセンス機能を含んでいる。キャッシュ・メモリの内容が維持されている（バッテリーの適切な動作によって）ことをバッテリー・サブシステムが示しているRDACは、ここでは「有効な」キャッシュ（または、単純に「良いバッテリー」）と呼ばれる。逆に、そのバッテリ

ー・サブシステムが、故障しているバッテリー・サブシステムのためにそのキャッシュ・メモリの内容が疑わしいことを示しているRDACは、ここでは「無効な」キャッシュ（または、単純に「不良バッテリー」）と呼ばれる。

【0025】本発明はRDACの各ペアの内部で動作するステート・マシーンとして表現される方法を含んでいる。その方法はそのRAIDサブシステムがホスト・コンピュータ・システムのI/O要求の処理に使えない初期化の期間を減らしながら、冗長キャッシュ・メモリ

【0026】同期化の方法のステート・マシーンー第1のコントローラ

図4および図5は本発明の方法で、RDACのペアの第1のRDACの内部での動作を示しているフローチャートである。このフローチャートは関連付けられているRDAC 118. 1および118. 2のCPU 112. 1および112. 2の内部で動作する方法の動作を記述している。初期化を実行する第1のRDACは共有されるリソース（すなわち、ソフトウェア実装のセマフォまたは共有レジスタまたは他の等価な電子回路）をロックするために、コントローラのペアの第1のコントローラとしてランダムなチャンスによって決定することができる。代わりに、RDACのペアのうちの第1のRDACはそのRAIDサブシステムの中の物理的な位置によって決定することができる。例えば、そのRDACペアが普通はそのRAIDサブシステムの中の共通のバックプレーン/バス・デバイスに挿入されるコントローラの各ペアの第1のコントローラがその場合最も低い番号のスロット（そのバックプレーン・デバイスのスロットが何らかの方法で識別されていると仮定して）、あるいは定義された位置により近い（例えば、電源により近い）スロットにあるRDACとして定義することができる。そのような電子回路の設計の選択は電子回路設計技術の分野に普通に習熟している人にはよく知られている。従って、この分野の技術に普通に習熟している人は初期化するために第1のRDACによって実行される図4および図5の処理が物理的に両方のRDAC（118. 1および118. 2）の中に存在し得ることが分かる。次の説明を簡単にするために、RDAC 118. 1がそのスタートアップの初期化を実行するための第1のコントローラであり、RDAC 118. 2が第2のコントローラであると仮定される。

【0027】図4および図5に示されているように、そして以下に説明されるように、RDAC 118. 1が初期化する時、その処理はミラー型のキャッシュ動作がその記憶サブシステム構成によってイネーブルされているかどうかを判定するための「チェック・イン」から始まる。そのRAIDサブシステムがミラー型のキャッシュ動作をするように構成されていないと第1のRDAC

118. 1が判定した場合、RDAC 118. 1は非ミラー型のキャッシュ・モードで動作を継続する。非ミラー型の動作は本発明に関してはこれ以上関心の対象となるものではなく、このモデルの完全性を示すためのものである。

【0028】ミラー型の動作がイネーブルされている場合、RDAC 118. 1および118. 2のキャッシュ116. 1および116. 2はミラー型の動作が開始される前に初期化されなければならない。そのミラー型のキャッシュはここで使われているように、2つのキャッシュ116. 1および116. 2のうちのどれか1つがフォーレン状態を示しているか、あるいは2つのRDAC 118. 1および118. 2のうちのいずれか1つが不良バッテリー（無効なキャッシュ内容）を示している場合に、同期化されていない。ネイティブ状態であって、バッテリーのステータスが「良」であるRDACはここでは「使用可能」と呼ばれる。一方、ステータスが不良またはフォーレンのいずれかであるRDACは、ここでは「使用不可能」と呼ばれる。2つのミラー型のキャッシュのうち1つだけが使用可能である場合、他のキャッシュは使用可能キャッシュから更新してそのミラー型のキャッシュを同期化させることができる。RDAC 118. 1および118. 2が両方とも使用不可能であった場合、そのミラー型のキャッシュ116. 1および116. 2はRDAC 118. 1および118. 2のいずれかの中にある既存のデータを使って同期化することができない。代わりに、キャッシュ・メモリは両方のキャッシュ116. 1および116. 2をクリアし、新しく初期化されたキャッシュの内容でミラー型のキャッシュ動作を再スタートすることによって同期化される。両方のキャッシュ・メモリ116. 1および116. 2がバージされてキャッシングが再スタートされた時、そして両方のキャッシュ・メモリのバッテリーが良好である時、そのキャッシュは同期化されていると言われ、そして両方とも現在のRAIDサブシステムに対してネイティブとなる。同様に、両方のキャッシュ・メモリが使用可能である場合（例えば、バッテリーが「良」であって両方のキャッシュ・メモリが現在のRAIDサブシステムに対してネイティブである場合、そのキャッシュ・メモリは同期化されていると言われる。他のすべての状態においては、2つのキャッシュ・メモリ116. 1および116. 2のうちの1つがRDAC 118. 1および118. 2の1つによって使われ、その有効な、ネイティブ・キャッシュ・メモリの内容が無効な、あるいはフォーレンのキャッシュ・メモリに対してコピーされる。そのコピー（および管理データ構造およびフラグの関連付けられた更新）のプロセスは2つのキャッシュ・メモリを同期化させるように働き、そしてそれらを両方とも現在のRAIDサブシステムに対してネイティブとする。

【0029】比較的大きな記憶サブシステムに対するキャッシュ・コントローラにおいては、2レベルのキャッシュ・メモリ、限定された量のデータにアクセスするための不揮発性メモリ、および比較的大きな量のデータにアクセスするための揮発性メモリを利用するのが普通である。例えば、組み合わせられたキャッシュ・メモリの使用を制御するための或る種の大型のデータ構造を利用して、そのデータに対する迅速なアクセスを提供することができる。そのようなキャッシュ制御ブロック（CCB）は比較的大きくなる可能性があるが、それらはキャッシュ・メモリのサブシステムの中に記憶されているデータに対する迅速なアクセスを提供する。実際のキャッシュされるデータおよびより小さな制御ブロックはキャッシュ・メモリ・サブシステムの揮発性部分に格納される。これらの回復制御ブロック（RCB）はキャッシュ・メモリ・サブシステムの揮発性部分の中のCCBを復元する（すなわち、再構築する）ために使われる。1つのRDACが代替のRDACのキャッシュ内容をコピーする時、その代替のRDACの中の揮発性のキャッシュ・メモリの部分だけをコピーする必要がある。揮発性の部分（CCB）はコピーされたRCBおよび実際のキャッシュ・データへの参照によって再構築される。

【0030】図4の要素400はミラー型のキャッシュ・モードがイネーブルされているかどうかを判定するようにRDAC 118. 1の内部で動作する。ミラー型のキャッシュ・モードがイネーブルされていないと要素400が判定した場合、要素402は非ミラー型のモードでその記憶サブシステムにおいてI/O要求を処理することができる。このモードは本発明にとってはこれ以上関心の対象とはならないので、これ以上は説明されない。ミラー型のキャッシュ・モードの動作が必要であると要素400が判定した場合、処理は要素404へ継続して代替のRDAC（第2のRDAC 118. 2）のステータスを判定する。

【0031】要素404が、代替のRDAC 118. 2がチェックインした（すなわち、第1のRDAC 118. 1との動作を同期化するための処理に対する準備ができている）と判定した場合、処理は要素406へ継続する。要素406はいずれかのRDAC（118. 1または118. 2）が使用可能である（すなわち、ステータスが「ネイティブ」および「良いバッテリー」である）かどうかを判定する。いずれかのRDACが使用可能でない場合、処理は図5のラベル「E」へ継続し、処理を完了し、キャッシュ・メモリ・サブシステムをチャージすることによってミラー型の動作に入る。さもなければ、要素408が動作して第1のRDAC 118. 1のキャッシュ・メモリ・サブシステムが使用可能であるかどうかを判定する。（少なくとも）RDAC 118. 1のステータスが「使用可能」である場合、その処理は図5のラベル「B」へ継続し、ミラー型のモードの

動作が開始される。さもなければ、処理は図5のラベル「A」において継続し、使用可能な第2のRDAC 118. 2のキャッシュの内容がコピーされる。

【0032】図4の要素404が、代替のRDAC 118. 2がまだチェック・インしていない（すなわち、第1のRDAC 118. 1との同期化を開始するために十分には初期化されていない）と判定した場合、要素410が、第1のRDAC 118. 1のキャッシュ・メモリ・サブシステムが使用可能であるかどうかを次に判定する。第1のRDACのキャッシュ・メモリ・サブシステムが使用可能であった場合、要素416および418はその代替のRDAC 118. 2に対して5秒間待機してチェック・インすることを繰り返す行なう。特に要素416は代替のRDAC 118. 2がチェック・インしたかどうかを判定する。チェック・インしていた場合、処理は図5のラベル「B」において継続する。チェック・インしていなかった場合、処理は要素418において継続し、5秒のタイムアウト値が過ぎたかどうかをテストする。そうであった場合、処理は図5のラベル「C」において継続し、代替のRDACのレイト・チェック・インを待ちながら、第1のRDAC 118. 1のライトスルー動作を開始する。5秒のタイムアウト期間が経過していなかった場合、処理は要素416へループバックすることによって継続する。

【0033】上記の要素410の動作が、第1のRDAC 118. 1のキャッシュ・メモリ・サブシステムが使用可能でないと判定した場合、処理は要素412および414に継続して代替のRDAC 118. 2のレイト・チェック・インを待つ。特に、要素412は、代替のRDAC 118. 2がチェック・インしたかどうかを判定する。そうである場合、処理は図5のラベル

「A」において継続し、1つのキャッシュ・メモリの内容を他のキャッシュ・メモリへコピーすることによって処理を完了する。代替のRDAC 118. 2がチェック・インしていなかった場合、処理は要素414において継続し、長いタイムアウト期間（45秒）が経過したかどうかを判定する。第1のRDAC 118. 1のキャッシュはその現在の状態において使用不可能なので、代替のRDACのチェック・インに対する待機が要素416および418に関して上記の5秒間の短い待機を超えて延長される。その延長されたタイムアウト期間が経過した場合、処理は図5のラベル「E」において継続する。それ以外の場合、処理は要素412へループバックすることによって継続する。

【0034】この分野の技術に普通に習熟している人には要素412～418の処理において任意のタイムアウト期間の値を適用できることが分かる。そのタイムアウトは第1のRDACができるだけ早くライトスルー動作を開始できるように意図的に比較的短く設定されている。その記憶サブシステムは第1のRDAC 118.

1をライトスルー・モードで処理することによって動作できるようになる。上記の要素416～418における短いタイムアウト期間（約5秒が望ましい）によって、その記憶サブシステムはRDACペアの1つだけが完全に動作できるようになる。上記の要素412～414に関して延長された時間（45秒が好ましい）によって、第2のRDACの追加の時間が、例えば、オペレータによって簡単に修理され得る任意の遅延時間をカバーする。後で第2のRDACがチェック・インする時、その冗長ペアは完全に同期化され、ミラー型された動作を開始することができる。

【0035】ライトスルーのキャッシュ・モードにおいて、すべてのI/O要求は第1のRDAC 118. 1に関連付けられているキャッシュ・メモリ116. 1の使用または変更をバイパスするような方法で実行される。言い換えれば、書込みのI/O要求はRAIDサブシステムのディスク・アレー108に対して直ちにポストされる。読出しの要求だけがキャッシュ116. 1の内容を更新することになる。ライトスルー・モードでのRAIDコントローラの動作によって、キャッシュ116. 1の中には新しい汚れた（dirty）データ（ライトバックのキャッシュ・モードで生成されるような、遅延されたポスティングを待っているデータ）は生成されない。デュアル・コントローラによる従来の設計と違って、本発明の方法および構造は、デュアル・コントローラのキャッシュが同期化されている間、ライトスルー・モードにおいて性能を落とした動作を許す。これに対して、従来の設計では冗長のコントローラのキャッシュの初期化/同期化を待つために、RAIDサブシステムのすべての動作を停止していた。RAIDサブシステムの性能はライトスルーのキャッシュ動作においては劣化するが、この劣化した性能は多くのRAIDのアプリケーションにおいて動作しないよりは好ましい。ここで図5を参照すると、図4のフローチャートは代替のRDAC 118. 2がキャッシュ・メモリの内容に関して使用可能であり、第1のRDAC 118. 1が使用不可能であることを判定したことに応答して、要素420のラベル「A」での動作を継続する。代替のRDAC 118. 2のRCBおよびキャッシュ・メモリ116. 2の内容が第1のRDAC 118. 1のキャッシュ・メモリ116. 1へコピーされる。上記のように、キャッシュ・メモリの不揮発性の部分の内容だけをコピーすれば済む。キャッシュ・メモリ116. 1の対応している揮発性の部分の内容（すなわち、CCB）はそのコピーされた不揮発性のキャッシュ・データから再構築することができる。次に、要素422がその代替キャッシュの内容のコピーが正常に行なわれたどうかを判定する。そのコピーが正常に行なわれた場合、動作は図5のラベル「B」において継続し、冗長キャッシュを同期化するプロセスを完了する。そのコピーがキャッシュ・メモリの

内容を正しく同期化するのに失敗した場合、処理は図5のラベル「E」において継続する。コピー・プロセスの成功または失敗はキャッシュ・メモリ間の情報の物理的な交換に関連している結果のステータス（例えば、物理的ステータス）によって判定される。

【0036】図4のフローチャートは結果のキャッシュを同期化するプロセスを完了するための処理を図5のラベル「B」において継続する。次に、要素426はすべての汚れたデータを同期化されたキャッシュからディスク・アレーへフラッシュすることによって、すべてのキャッシュ・データ構造およびリソースを再度要求する。次に、処理は要素428において継続し、完全にミラー型のモード動作に対して同期化されているRDACペアでのミラー型のライトバック・キャッシュ・モードへ進む。

【0037】図4のフローチャートは図5のラベル「C」において継続し、処理は代替のRDAC 118. 2の遅延されたチェック・インを待ちながら、第1のRDAC 118. 1によってライトスルー・モードの動作へ入る。要素430は第1のRDAC 118. 1のキャッシュ・メモリ116. 1からディスク・アレーへすべての汚れたデータをフラッシュすることによって、すべてのキャッシュ・データ構造およびリソースを再度要求する。次に処理は要素432から継続し、第1のRDAC 118. 1に対するキャッシュ・コヒーレンシー（coherency）フラグをクリアする。そのフラグをクリアすることは、第1のRDAC 118. 1が代替のRDAC 118. 2の遅延されたチェック・インを待ちながらライトスルー・モードで動作していることを代替のRDAC 118. 2に対して示す。そのコヒーレンシー・フラグは以下に説明されるように、第1のRDAC 118. 1がライトスルー・モードで動作中に故障し、そして第2のRDAC 118. 2がその故障したRDAC 118. 1に取って代わるのに十分な点まで実質的に初期化する場合に、正しい復元を確保するために使われる。そのコヒーレンシー・フラグが一旦クリアされると、処理は要素434および436について継続し、代替のRDAC 118. 2のレイト・チェック・インを待ちながら、ライトスルー・モードでI/O要求を処理する。特に、要素434は待機中の書込み要求があればそれを処理し、キャッシュ・メモリ116. 1をバイパスしてディスク・アレーに対して直接にそれらをポストする。次に、要素436はその代替のRDAC 118. 2が最終的にチェック・インしたかどうかをテストする。チェック・インしていなかった場合、処理は要素434へループバックすることによって継続する。チェック・インしていた場合、処理は要素438において継続し、第1のRDAC 118. 1のコヒーレンシー・フラグをセットしてから、処理は要素428において継続し、ミラー型のモードの動作に対し

て完全に同期化されたRDACペアでのミラー型のライト・バック・キャッシュ・モードへ進む。

【0038】図4のフローチャートは図5のラベル「E」において継続し、誤り状態を発生し（付属のコンピュータ・システムに対して実質的に知らせるために）、そして次にキャッシュ・メモリ116. 1の内容をパージする。RDACのどれかが他のRDACとの同期化の目的に使えない場合、両方のRDACはそれぞれのキャッシュ・メモリをパージし（そしてそれぞれの不揮発性メモリを維持するためにバッテリー・システムの適切なチャージングを待ち）、そしてミラー型のモードで動作を継続する。キャッシュ・メモリのペアは両方のキャッシュ・メモリのパージングによって同期化される。

【0039】＜同期化方法のステート・マシーン＞第2のコントローラ>図6はスタートアップの初期化を実行している第2のRDAC 118. 2の中での本発明の動作の方法のフローチャートである。第2のRDAC 118. 2は第1のRDAC 118. 1の動作の上記の説明のコンテキストにおいては「代替のRDAC」とも呼ばれている。上記のように、ここで使われている「代替のRDAC」という用語は、説明されている冗長ペアの特定のRDACに関係しているものである。第2のRDAC 118. 2の中で動作する本発明の方法を記述している図6に対する参照においては、「代替の」RDACは第1のRDAC 118. 1である。

【0040】図6に示されているように、要素600はまず記憶サブシステムがミラー型のモードの動作をイネーブルするように構成されているかどうかを判定する。ミラー型の動作がイネーブルされていなかった場合、処理は要素602において継続し、すべてのI/O要求を非ミラー型のモードで処理する。非ミラー型のモードの動作は本発明の関心の対象とはならないので、これ以上説明される必要はない。ミラー型の動作がイネーブルされていた場合、処理は要素604から継続し、いずれかのRDACがそのキャッシュ・メモリの内容に関して使用可能であるかどうかを判定する。上記のように、RDACはそのキャッシュ・メモリが良いバッテリー・サブシステムによって不揮発状態に保たれている場合に使用可能であり、そのRDACはそれが現在動作中の記憶サブシステムに関してネイティブであるとして識別される。いずれのRDACもそのキャッシュ・メモリの内容を他のキャッシュ・メモリへコピーするために使えない場合、処理は要素612において継続する。それ以外の場合、2つのRDACのうちの少なくとも1つがそのキャッシュ・メモリの内容に関して使用可能である場合、処理は要素606から継続する。

【0041】要素606は少なくとも第2のRDACのキャッシュ・メモリが使用可能であるかどうかを判定す

る。第2のRDACが使用可能でない場合、処理は要素608から継続して代替（第1の）RDAC 118. 1の内容を第2のRDAC 118. 2のキャッシュ・メモリ116. 2へコピーする。上記のように、キャッシュ・メモリの不揮発性の部分に格納されているキャッシュ・メモリの部分だけをコピーすれば済む。揮発性のメモリの部分に格納されているキャッシュ・メモリの管理に関連している他のデータ構造は、不揮発性部分の中にコピーされた情報から再構築することができる。次に、要素610はそのコピー動作が正常に行なわれたかどうかを判定する。そのコピー・プロセスの成否はキャッシュ・メモリ間での情報の物理的な交換に付随している結果のステータス（例えば、物理的ステータス）によって判定される。第1のRDAC 118. 1からのキャッシュの複製している部分においてコピーが成功した場合、処理は要素616から継続する。コピーが失敗した場合、処理は要素612から継続して誤り状態をセットし（付属のコンピュータ・システムに対して実質的に知らせ）、次にキャッシュの内容をパージすることによって第2のRDAC 118. 2のキャッシュ・メモリ116. 2を同期化する。処理は次に要素614から継続し、ライトバック型のミラー型キャッシュ動作を開始する。

【0042】要素606が、第2のRDAC 118. 2のキャッシュ116. 2が使用可能であると判定した場合、あるいは要素610が代替（第1の）RDAC 118. 1からのキャッシュのコピーが成功であった（従って、冗長キャッシュの同期化が成功した）と判定した場合、処理は要素616から継続してその複製の冗長キャッシュに対する同期化処理を完了する。特に、要素616は現在代替（第1の）116. 1に同期化されているキャッシュ116. 2の不揮発性部分からすべての揮発性管理データ構造（例えば、CCB）を復元する。次に処理は要素614から継続し、上記のように、ライトバックのミラー型キャッシュ・モードにおいてI/O要求の処理を開始する。

【0043】デュアル・キャッシュの同期化に対応する図4～図6を参照しての上記の処理は、2つのキャッシュ116. 1または116. 2のうちのどれが（もしあれば）キャッシュを同期化するために他のキャッシュへコピーされるかを本質的に決定する。代わりに、その処理はいずれかのキャッシュが使用可能でないと判定し、両方のキャッシュの内容をパージすることによってキャッシュを同期化させることができる。次の表1はデュアルRDAC 118. 1および118. 2の可能な各状態の下でのキャッシュの同期化のために取られるべき対策を要約している。

【0044】

【表1】

RDAC1のステータス	RDAC1のバッテリー	RDAC2のステータス	RDAC2のバッテリー	対策
ネイティブ	良	ネイティブ	良	なし
ネイティブ	良	ネイティブ	不良	1->2
ネイティブ	良	フォーレン	良	1->2
ネイティブ	良	フォーレン	不良	1->2
ネイティブ	不良	ネイティブ	良	2->1
ネイティブ	不良	ネイティブ	不良	バージ
ネイティブ	不良	フォーレン	良	バージ
ネイティブ	不良	フォーレン	不良	バージ
フォーレン	良	ネイティブ	良	2->1
フォーレン	良	ネイティブ	不良	バージ
フォーレン	良	フォーレン	良	バージ
フォーレン	良	フォーレン	不良	バージ
フォーレン	不良	ネイティブ	良	2->1
フォーレン	不良	ネイティブ	不良	バージ
フォーレン	不良	フォーレン	良	バージ
フォーレン	不良	フォーレン	不良	バージ

キャッシュの同期化のアクション

上の表1の中で、アクション「1->2」は第1のRDACキャッシュ116. 1から第2のRDACキャッシュ116. 2へのコピーを示し、「2->1」は第2のRDACキャッシュ116. 2から第1のRDACキャッシュ116. 1へのコピーを示し、そして「バージ」はキャッシュ116. 1および116. 2の両方のページを示す。キャッシュ116. 1および116. 2の同期化および「不良」の指示をリセットするために両方のバッテリー・サブシステムのチャージングに続いて、RDAC 118. 1および118. 2の両方のステータスが「ネイティブ」に変更される。

【0045】<キャッシュのコヒーレンシー・フラグ>故障したRDACのテイクオーバー>1つのキャッシュ・コヒーレンシー・フラグがRDACのペアのそれぞれに関連付けられている。キャッシュ・コヒーレンシー・フラグはクリアされている場合、その対応しているRDAC 118. 1がライトスルー・モードで動作していたこと、および、従ってそのキャッシュがディスクへフラッシュされたことを示す。RDAC 118. 1の動作のこの状態については、図5の要素430~438に関して説明されている。キャッシュのコヒーレンシー・フラグがセットされた場合、その対応しているRDAC 118. 1はその時点ではライトスルー・モードで動作していなかったことを示す。ライトスルー・モードで動作しているRDAC 118. 1のキャッシュ・メモリ116. 1はバージされる（ディスクへフラッシュされる）。ライトスルー・モードで動作していない時、RDAC 118. 1のキャッシュ・メモリ116. 1にはディスクへまだポストされていないデータ（汚れたデータ）が含まれている可能性がある。

【0046】RDAC 118. 1がライトスルー・モ

ードで動作して（代替のRDAC 118. 2の遅延されたチェック・インを待っていて）、ライトスルー・モードで動作している間に故障した場合、そのディスク・アレーはすべてのディスクの動作がそのディスク・アレーに対して直接にキャッシュ116. 1を通じて書き込まれたので、コンシステント状態にあることが知られている。実質的に、代替のRDAC 118. 2は処理中にそのチェック・インを完了することができ、そして第1のRDAC 118. 1の状態を決定するためにシークすることができる。そのような環境における代替のコントロール118. 2はディスク・アレーがコンシステント状態にあることを、故障したRDAC 118. 1のコヒーレンシー・フラグがクリアされたことから検出することができる。次に、代替のRDAC 118. 2は自分のキャッシュ・メモリ116. 2をクリアし、その処理をライトスルー・モードで開始する。本発明のこの状態によってデータの記憶およびキャッシュ・メモリの完全性が、動作の継続を許しながら、ペアのRDACのうちの1つが全く故障した場合においても確保される。

【0047】本発明は図面の詳細において、そして前記の説明の中で示されてきたが、そのような図による表現と説明は例としてみなされるべきであり、性格的に制限をするものではない。好ましい実施例およびそれからの小変形だけが示されて説明されてきたこと、および本発明の精神の範囲内にあるすべての変更および修正は保護されるのが望ましいことを理解されたい。

【図面の簡単な説明】

【図1】 本発明の構造および方法が適用できる代表的なRAID記憶サブシステムのブロック図である。

【図2】 デュアル・アクティブ動作モードに構成され

ている図1のRDACを示しているブロック図である。

【図3】 アクティブ/パッシブのペア動作モードに構成されている図1のRDACを示しているブロック図である。

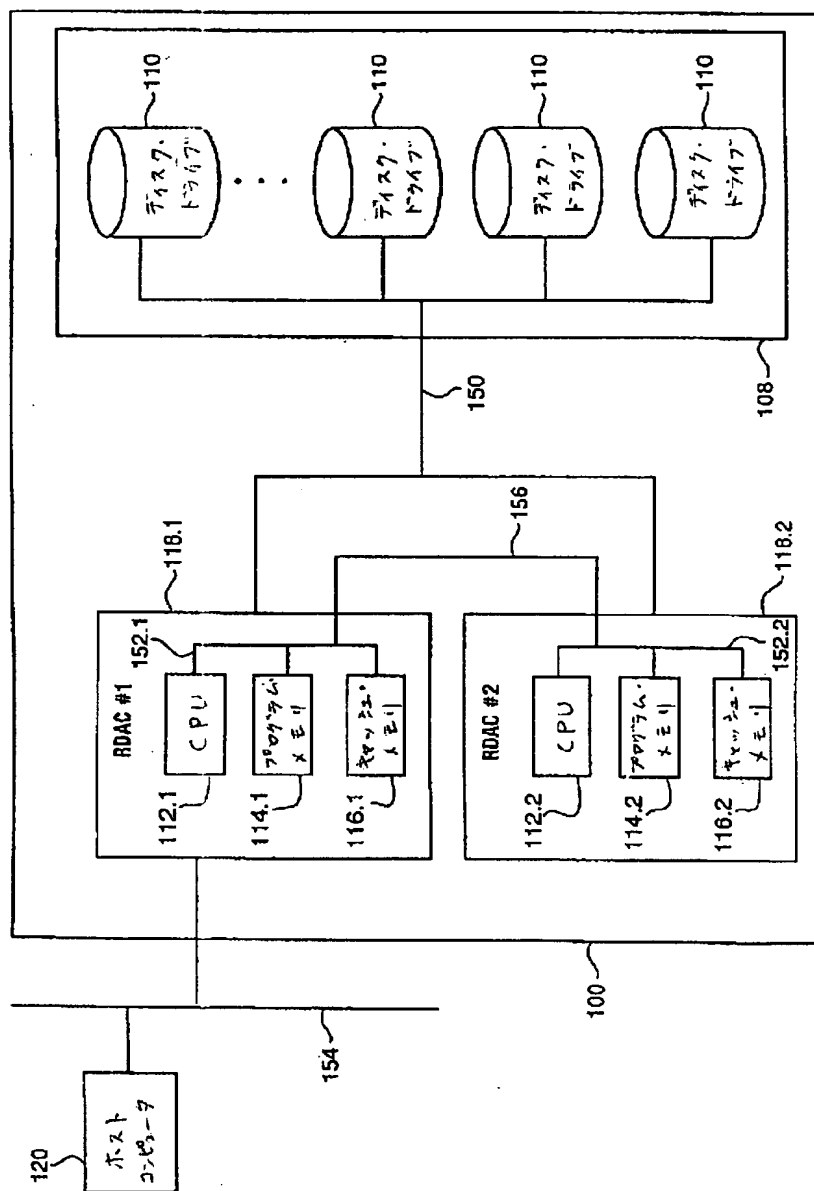
【図4】 RDACの第1のペアのチェック・インの動作を示しているフローチャートである。

作を示しているフローチャートである。

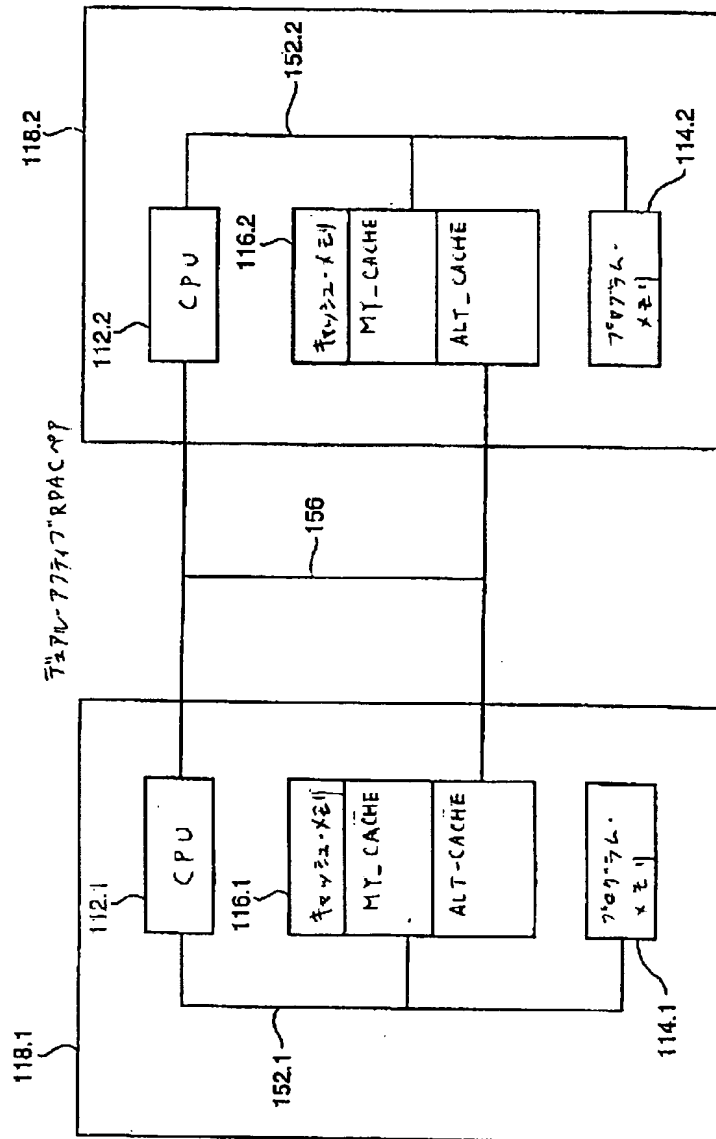
【図5】 RDACの第1のペアのチェック・インの動作を示しているフローチャートである。

【図6】 RDACの第2のペアのチェック・インの動作を示しているフローチャートである。

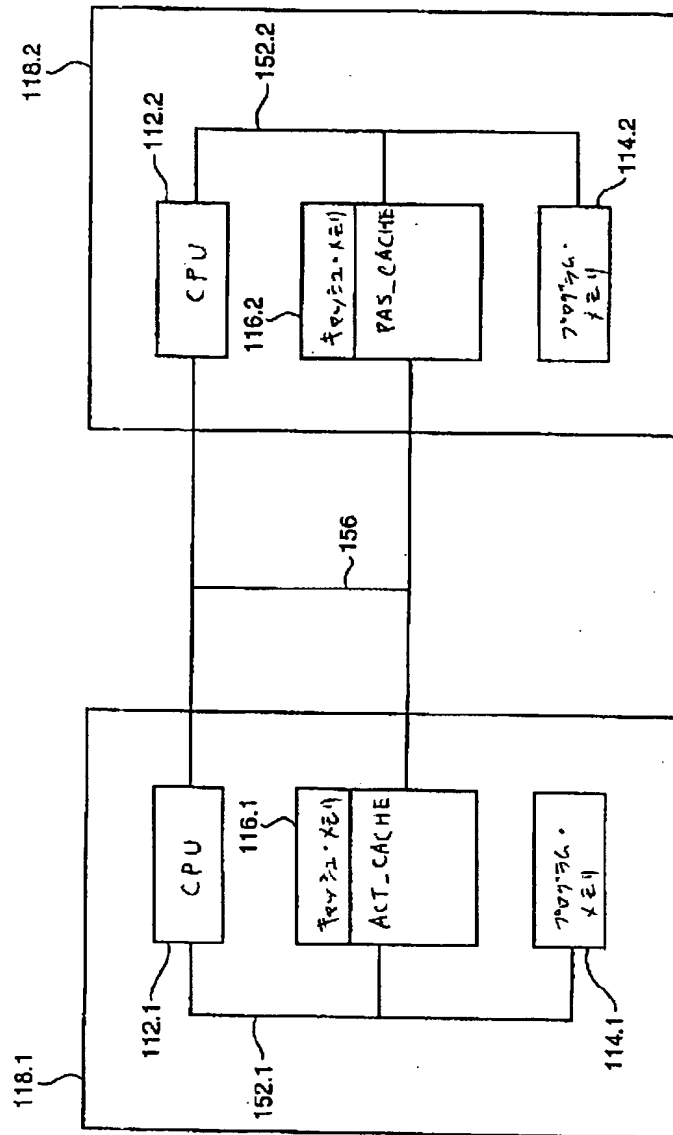
【図1】



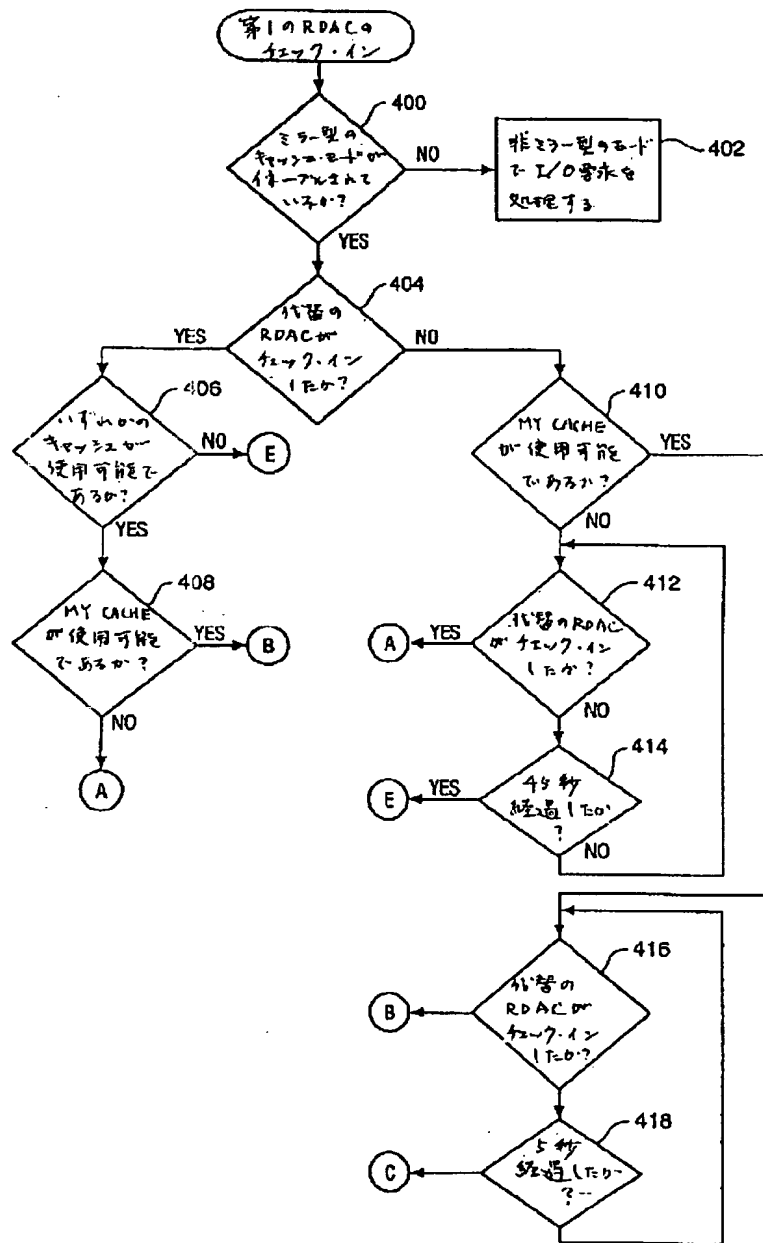
【図2】



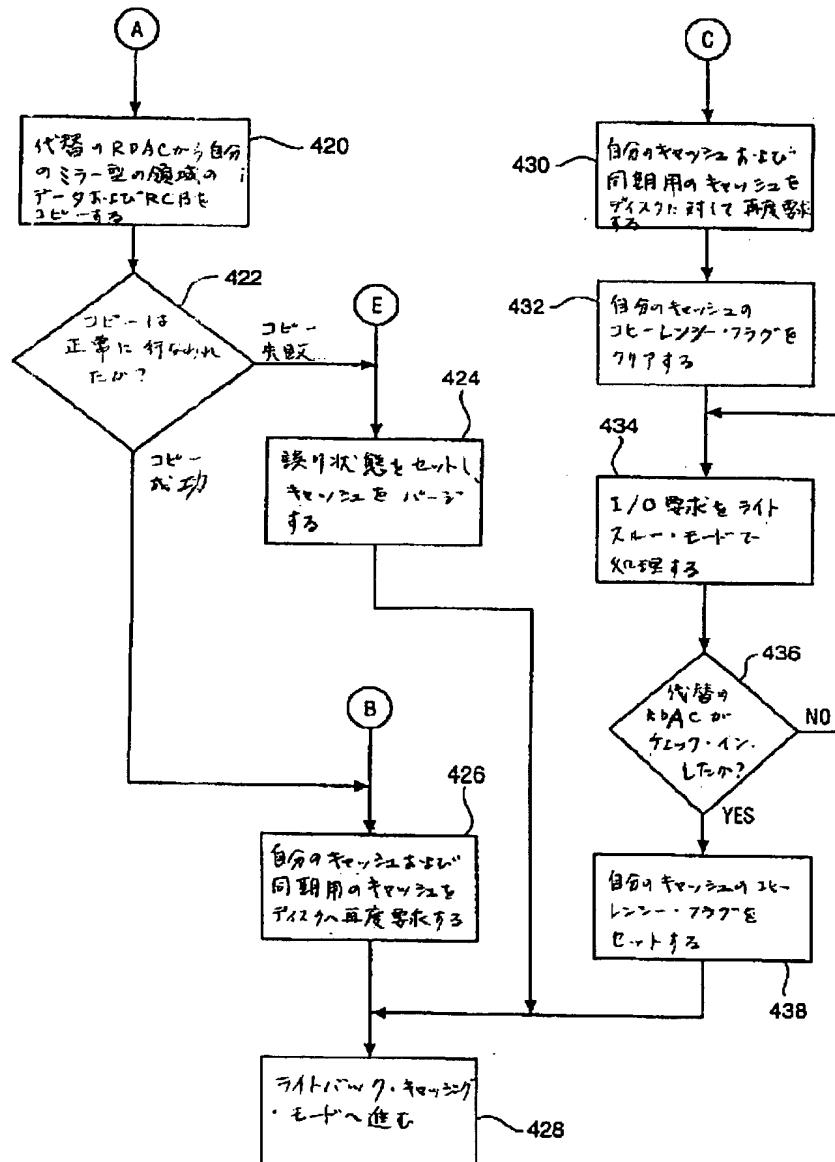
【図3】



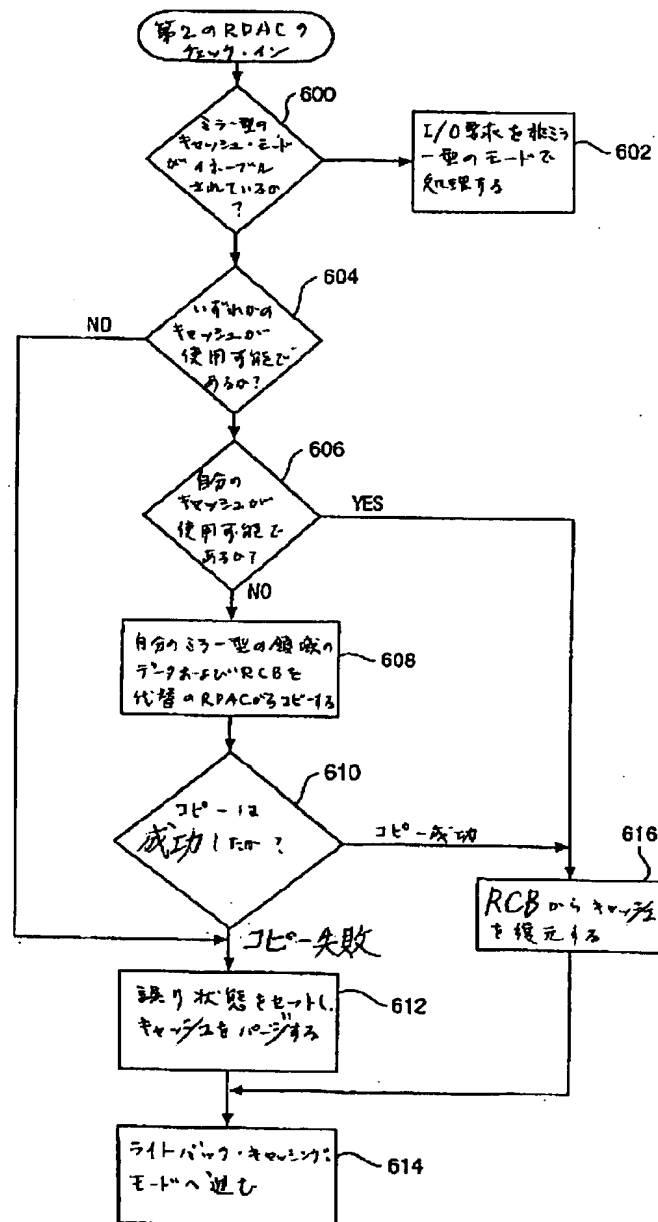
【図4】



【図5】



【図6】



フロントページの続き

(72)発明者 マックス エル. ジョンソン
アメリカ合衆国 カンザス州 67226 ウ
ィチタ、スイートベイ サークル 4110